UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/646,282	08/21/2003	Frank Liebenow	ACER-45269	9378

116 7590 05/06/2009

PEARNE & GORDON LLP
1801 EAST 9TH STREET
SUITE 1200
CLEVELAND, OH 44114-3108

EXAMINER	
DAO, THUY CHAN	

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
05/06/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/646,282	LIEBENOW, FRANK	
	Examiner	Art Unit	
	Thuy Dao	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 January 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-24,26-42,44-51 and 54-70* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-24,26-42,44-51 and 54-70* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 August 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on January 23, 2009.

2. Claims 1-4, 6-24, 26-42, 44-51, and 53-70 have been examined.

Response to Amendments

3. In the instant amendment, claims 1, 6, 11, 15, 16, 21, 26, 34, 35, 39, 44, 48, 53, and 64 have been amended.

4. The objection to claim 21 is withdrawn in view of Applicant's amendments.

Response to Arguments

5. Applicants' arguments regarding claims 1-4, 6-24, 26-29, 39-42, 44-51, and 53-63 have been considered. The examiner notes that applicants' amendments necessitated the new ground of rejection presented in this Office action. Thus, applicants' arguments are moot in view of the new ground of rejection.

6. Applicants' arguments regarding claims 30-38 and claims 64-70 have been considered but are not persuasive.

Claim 30 (Remarks, page 17)**:**

The Applicants asserted,

> "The Office Action stated that Warnes discloses limiting the use of one or more of the plurality of registers. Applicant disagrees with the Examiner's interpretation of the cited reference. Upon a close review, the cited passage (col. 11, Table 3, col. 11::14-col. 12:61), it seems that Warnes discloses encoding of registers (Table 3) for faster and simpler instruction decoding (col. 12, lines 14-15), but not limiting number of registers available for use as claim 30."

The examiner respectfully disagrees with Applicants' assertions. Warnes explicitly teaches:

limiting the use of one or more of the plurality of registers (e.g., col.11, Table 2, limiting the use of 32-bit ISA registers R0-R3 and R13-16 and instead using 3-bit Compressed ISA registers 0-7; see further in Table 4 and col.12: 35-61)

based on the frequency of use of one or more of the plurality of registers (e.g., col.11, Table 2, col.11: 21-27, based on the most frequently used registers, which are identified as R0-R3 and R13-16).

Dependent claims 31-38 (Remarks, page 17)**:**

Dependent claims 31-38 are also rejected based on virtue of their dependencies on the rejected base claim 30.

Claim 64 (Remarks, page 19)**:**

The Applicants asserted,

> "Claim 64 has been amended to clarify that tuning an instruction set is by selecting an op-code representation from a plurality of pre-determined sets. As discussed with regard to the patentability of claim 11, Srinivasamurthy et al. does not provide a plurality of pre-determined op-codes to be selected when tuning an instruction set. Therefore, Applicant submits Srinivasamurthy et al. cannot anticipate amended claim 64 and an early notification of allowance thereof is respectfully solicited."

The examiner respectfully disagrees with Applicants' assertions. Warnes explicitly teaches *means for tuning an instruction set by selecting an op-code representation from a plurality of pre-determined sets to an instruction* (e.g. FIG. 4b, at least 4 pre-determined sets sEcopcode 1-4 and selecting a specific sEcopcode based on the associated instruction sequences also illustrated in FIG. 4b).

Dependent claims 65-70 (Remarks, page 20)**:**

Dependent claims 65-70 are also rejected based on virtue of their dependencies on the rejected base claim 64.

Claim Rejections – 35 USC §102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 30-33 and 36-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Warnes (art of record, US Patent No. 7,051,189).

Claim 30:

Warnes discloses *a method for optimizing the representation of a code sequence, comprising:*

determining the frequency of use of one or more registers within a plurality registers by the operations performed in the code sequence (e.g., FIG. 1, blocks 102-106, col.8: 56 – col.9: 38)*;*

limiting the use of one or more of the plurality of registers (e.g., col.11, Table 2, limiting the use of 32-bit ISA registers R0-R3 and R13-16 and instead using 3-bit Compressed ISA registers 0-7 as further illustrated in Table 4 and col.12: 35-61)

based on the frequency of use of one or more of the plurality of registers (e.g., col.11, Table 2, col.11: 21-27, based on the most frequently used registers, which are identified as R0-R3 and R13-16)*; and*

tuning the instruction set for assigning a target-code representation for one or more of the plurality of registers (e.g., col.11, Table 2, col.10: 51 – col.11: 27),

wherein the tuning of the instruction set is based on the frequency of use of the plurality of registers (e.g., col.9: 1-19; col.9: 61 – col.10: 27; col.12: 40 – col.13: 16).

Claim 31:

Warnes discloses *the method of claim 30, wherein the representation of a code sequence is a bit symbol representation* (e.g., col.7: 59 – col.8: 54).

Claim 32:

Warnes discloses *the method of claim 30, wherein the instruction set is a variable length instruction set* (e.g., col.7: 59 – col.8: 54).

Claim 33:

Warnes discloses *the method of claim 30, wherein the instruction set is a constant length instruction set* (e.g., col.10: 17 – col.11: 62).

Claim 36:

Warnes discloses *the method of claim 30, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed* (e.g., col.13: 3-64).

Claim 37:

Warnes discloses *the method of claim 36, wherein the representation of operation frequency is a frequency distribution* (e.g., col.7: 40 – col.8: 54).

Claim 38:

Warnes discloses *the method of claim 37, wherein the frequency distribution is a histogram* (e.g., col.9: 56 – col.10: 51).

9. Claims 64-70 are rejected under 35 U.S.C. 102(e) as being anticipated by Srinivasamurthy (art of record, US Patent Publication No. 2005/0028132 A1).

Claim 64:

Srinivasamurthy discloses *an optimized code generator for generating a source code stored on a computer readable medium and configured to be executed by a computer, comprising:*

means for scanning the code sequence to determine a static determining the frequency of operations performed in a code sequence (e.g., [0045]-[0046], [0049]-[0050])*;*

means for determining an executed frequency of operations for the code sequence (e.g., [0016]-[0018], [0053]-[0056])*; and*

means for tuning an instruction set for by selecting an op-code representation from a plurality of pre-determined sets to an instruction (e.g., [0021], [0030], FIG. 4b, at least 4 pre-determined sets sEcopcode 1-4 and selecting a specific sEcopcode based on the associated instruction sequences; and further in [0046], [0049], [0066]-[0068])*,*

wherein the tuning of the instruction set is based on the static frequency of operations and the executed frequency of operations (e.g., [0012], [0016]. [0050], [0056])*.*

Claim 65:

Srinivasamurthy discloses *the optimized code generator of claim 64, comprising a compiler* (e.g., [0016]-[0024], [0046]-[0051])*.*

Claim 66:

Srinivasamurthy discloses *the optimized code generator of claim 64, comprising an assembler* (e.g., [0035]-[0038], [0040]-[0045])*.*

Claim 67:

Srinivasamurthy discloses *the optimized code generator of claim 64, wherein the instruction set is a variable length instruction set* (e.g., FIG. 4b, [0052]-[0055]).

Claim 68:

Srinivasamurthy discloses *the optimized code generator of claim 64, wherein the instruction set is a constant length instruction set* (e.g., FIG. 3, [0046]-[0051]).

Claim 69:

Srinivasamurthy discloses *the optimized code generator of claim 64, wherein a representation of operations frequency is a frequency distribution* (e.g., [0017], [0041], [0067], [0075]).

Claim 70:

Srinivasamurthy discloses *the optimized code generator of claim 69, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

Claim Rejections – 35 USC §103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1-4, 7-15, 17-20, 39-42, 45-51 and 54-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Srinivasamurthy in view of US Patent Publication No. 2003/0041320 A1 to Sokolov (art made of record, herein "Sokolov").

Claim 1:

Srinivasamurthy discloses *a method for optimizing a representation of a code sequence, comprising:*

scanning the code sequence to determine a static frequency of operations in the code sequence (e.g., [0045]-[0046], [0049]-[0050])*;*

determining an executed frequency of operations for the code sequence (e.g., [0021]-[0022], [0039] and [0046])*; and*

tuning an instruction set for assigning an op-code representation to an instruction (e.g., [0021], [0030], [0046], [0049], [0066]-[0068])*,*

wherein the tuning of the instruction set is based on the static frequency of operations and the executed frequency of operations (e.g., [0012], [0016]. [0050], [0056])*.*

Srinivasamurthy does not explicitly disclose *[an executed frequency of operations] within multiple times executed loops.*

However, in an analogous art, Sokolov further discloses *an executed frequency of operations within multiple times executed loops* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 2:

Srinivasamurthy discloses *the method of claim 1, wherein the representation of a code sequence is a bit symbol representation* (e.g., [0024], [0032])*.*

Claim 3:

Srinivasamurthy discloses *the method of claim 1, wherein the instruction set is a variable length instruction set* (e.g., FIG. 3, [0046]-[0051])*.*

Claim 4:

Srinivasamurthy discloses *the method of claim 1, wherein the instruction set is a constant length instruction set* (e.g., FIG. 4b, [0052]-[0055]).

Claim 7:

Srinivasamurthy discloses *the method of claim 1, further comprising the step of providing a representation of operation frequency, which represents the frequency of operations performed* (e.g., [0021]-[0022], [0039]).

Claim 8:

Srinivasamurthy discloses *the method of claim 7, wherein the representation of operation frequency is a frequency distribution* (e.g., [0046], [0070]).

Claim 9:

Srinivasamurthy discloses *the method of claim 8, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

Claim 10:

Srinivasamurthy discloses *the method of claim 1, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set* (e.g., [0052]).

Claim 11:

Srinivasamurthy discloses *a method for optimizing the representation of a code sequence, comprising:*

determining the frequency of operations performed in the code sequence (e.g., [0045]-[0046], [0049]-[0050]; [0016]-[0018], [0053]-[0056]);

providing a plurality of pre-determined instruction sets with each pre-determined instruction set comprising instructions including assigned op-code representations (e.g., [0021], [0030], [0046], [0049], [0066]-[0068]);

selecting one of the plurality of predetermined instruction sets based on the determined frequency of operations performed (e.g., [0012], [0016]. [0050], [0056]).

Srinivasamurthy does not explicitly disclose *[each pre-determined instruction set] being optimized for a frequency of a particular operation.*

However, in an analogous art, Sokolov further discloses *each pre-determined instruction set being optimized for a frequency of a particular operation* (e.g., [0066]; FIG. 9, 12 and related text).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 12:

Srinivasamurthy discloses *the method of claim 11, wherein the representation of a code sequence is a bit symbol representation* (e.g., [0024], [0032]).

Claim 13:

Srinivasamurthy discloses *the method of claim 11, wherein the instruction set is a variable length instruction set* (e.g., FIG. 4b, [0052]-[0055]).

Claim 14:

Srinivasamurthy discloses *the method of claim 11, wherein the instruction set is a constant length instruction set* (e.g., FIG. 3, [0046]-[0051]).

Claim 15:

Srinivasamurthy does not explicitly disclose *the method of claim 11, wherein the step of determining operation frequency may further comprising determining an executed frequency of operations within multiple times executed loops for the code sequence.*

However, in an analogous art, Sokolov further discloses *determining an executed frequency of operations within multiple times executed loops for the code sequence* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 17:

Srinivasamurthy discloses *the method of claim 11, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed* (e.g., [0035]-[0038], [0065]-[0070]).

Claim 18:

Srinivasamurthy discloses *the method of claim 17, wherein the representation of operation frequency is a frequency distribution* (e.g., [0021]-[0022], [0039]).

Claim 19:

Srinivasamurthy discloses *the method of claim 18, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

Claim 20:

Srinivasamurthy discloses *the method of claim 11, wherein a more compact version of the code sequence is accomplished through shortening of bit symbol representation of an op-code of the instruction set* (e.g., [0052], [0058]-[0063]).

Claim 39:

Srinivasamurthy discloses *a computer readable medium upon which is stored computer readable code for optimizing a code sequence, wherein said computer readable code upon being executed on a computer causes steps comprising:*

scanning the code sequence to determine a static frequency of operations in the code sequence (e.g., [0045]-[0046], [0049]-[0050])*; and*

determining an executed frequency of operations for the code sequence (e.g., [0016]-[0018], [0053]-[0056])*;*

tuning an instruction set for assigning an op-code representation to an instruction (e.g., [0021], [0030], [0046], [0049], [0066]-[0068])*,*

wherein the tuning of the instruction set is based on the static frequency of operations and the executed frequency of operations (e.g., [0012], [0016]. [0050], [0056])*.*

Srinivasamurthy does not explicitly disclose *[an executed frequency of operations] within multiple times executed loops.*

However, in an analogous art, Sokolov further discloses *an executed frequency of operations within multiple times executed loops* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 40:

Srinivasamurthy discloses *the computer readable medium of claim 39, wherein the representation of a code sequence is a bit symbol representation* (e.g., [0024], [0032])*.*

Claim 41:

Srinivasamurthy discloses *the computer readable medium of claim 39, wherein the instruction set is a variable length instruction set* (e.g., FIG. 4b, [0052]-[0055])*.*

Claim 42:

Srinivasamurthy discloses *the computer readable medium of claim 39, wherein the instruction set is a constant length instruction set* (e.g., FIG. 3, [0046]-[0051])*.*

Claim 45:

Srinivasamurthy discloses *the computer readable medium of claim 39, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed* (e.g., [0017], [0041], [0067], [0075]).

Claim 46:

Srinivasamurthy discloses *the computer readable medium of claim 45, wherein the representation of operation frequency is a frequency distribution* (e.g., [0021]-[0022], [0039]).

Claim 47:

Srinivasamurthy discloses *the computer readable medium of claim 46, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

Claim 48:

Srinivasamurthy discloses *an optimized computing assembly, comprising:*

a processor coupled with a memory for executing programs; an optimized code generator operationally coupled with the processor and the memory (e.g., [0016]-[0024], [0035]-[0038]),

the optimized code generator for scanning the code sequence to determine a static frequency of operations in the code sequence (e.g., [0045]-[0046], [0049]-[0050]),

for determining an executed frequency of operations for the code sequence (e.g., [0016]-[0018], [0053]-[0056]), *and*

for tuning an instruction set for assigning an op-code representation selected according to the executed frequency of operations to an instruction (e.g., [0021], [0030], [0046], [0049], [0066]-[0068]),

wherein the tuning of the instruction set is based on the static frequency of operations and the executed frequency of operations (e.g., [0012], [0016]. [0050], [0056]).

Srinivasamurthy does not explicitly disclose *[an executed frequency of operations] within multiple times executed loops.*

However, in an analogous art, Sokolov further discloses *an executed frequency of operations within multiple times executed loops* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 49:

Srinivasamurthy *the optimized computing assembly of claim 48, wherein the representation of a code sequence is a bit symbol representation* (e.g., [0024], [0032]).

Claim 50:

Srinivasamurthy discloses *the optimized computing assembly of claim 48, wherein the instruction set is a variable length instruction set* (e.g., FIG. 4b, [0052]-[0055]).

Claim 51:

Srinivasamurthy discloses *the optimized computing assembly of claim 48, wherein the instruction set is a constant length instruction set* (e.g., FIG. 3, [0046]-[0051]).

Claim 54:

Srinivasamurthy discloses *the optimized computing assembly of claim 48, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed* (e.g., [0017], [0041], [0067], [0075]).

Claim 55:

Srinivasamurthy discloses *the optimized computing assembly of claim 54, wherein the representation of operation frequency is a frequency distribution* (e.g., [0021]-[0022], [0039]).

Claim 56:

Srinivasamurthy discloses *the optimized computing assembly of claim 55, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

Claim 57:

Srinivasamurthy discloses *an optimized code generator for generating source code stored on a computer readable medium and configured to be executed by a computer, comprising:*

a read executable for reading the source code; a translation executable operationally coupled with the read executable, the translation executable for translating the source code to an intermediate code (e.g., [0016]-[0024], [0040]-[0045]);

a scanning executable operationally coupled with the translation executable, the scanning executable for determining a static frequency of operations and an executed frequency of operations performed by the source code (e.g., [0045]-[0046], [0049]-[0050]) *and*

providing a representation based on the static frequency of operations and the executed frequency of operations (e.g., [0016]-[0018], [0053]-[0056]);

an optimizing translation executable operationally coupled with the scanning executable, the optimizing translation executable for translating the intermediate code to an object code including an optimized instruction set based on the static frequency of operations and the executed frequency of operations (e.g., [0021], [0030], [0046], [0049], [0066]-[0068]); *and*

a write executable operationally coupled with the optimizing translation executable, the write executable for outputting an optimized object code (e.g., [0012], [0016]. [0050], [0056]).

Srinivasamurthy does not explicitly disclose *[an executed frequency of operations] by analyzing loops.*

However, in an analogous art, Sokolov further discloses *an executed frequency of operations by analyzing loops* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 58:

Srinivasamurthy discloses *the optimized code generator of claim 57, comprising a compiler* (e.g., [0035]-[0038], [0040]-[0045]).

Claim 59:

Srinivasamurthy discloses *the optimized code generator of claim 57, comprising an assembler* (e.g., [0016]-[0024], [0046]-[0051]).

Claim 60:

Srinivasamurthy discloses *the optimized code generator of claim 57, wherein the instruction set is a variable length instruction set* (e.g., FIG. 4b, [0052]-[0055]).

Claim 61:

Srinivasamurthy discloses *the optimized code generator of claim 57, wherein the instruction set is a constant length instruction set* (e.g., FIG. 3, [0046]-[0051]).

Claim 62:

Srinivasamurthy discloses *the optimized code generator of claim 57, wherein the representation of operation frequency is a frequency distribution* (e.g., [0017], [0041], [0067], [0075]).

Claim 63:

Srinivasamurthy discloses *the optimized code generator of claim 62, wherein the frequency distribution is a histogram* (e.g., [0053], [0056]).

12. Claims 6, 16, 44, and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Srinivasamurthy in view of Sokolov and US Patent No. 6,718,539 to Cohen et al. (art made of record, herein "Cohen").

Claim 6:

Srinivasamurthy does not explicitly disclose *the method of claim 1, wherein the optimization of the code sequence may be executed by a microcode loadable from an external source.*

However, in an analogous art, *the optimization of the code sequence may be executed by a microcode loadable from an external source* (e.g., FIG. 3, col.8: 22-45 and col.9: 47 – col.10: 3, Microcode Memory 210, Sequence Optimization Unit 218, external sources, storing microcode to support optimization for processor/JVM).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Cohen's teaching into Srinivasamurthy and Sokolov's teaching. One would have been motivated to do so to support instruction set translation as suggested by Cohen (e.g., col.9: 47 – col.10: 3).

Claim 16:

The rejection of claim 11 is incorporated. Claim 16, which recite(s) the same limitations as those of claim 6, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 16.

Claim 44:

The rejection of claim 39 is incorporated. Claim 44, which recite(s) the same limitations as those of claim 6, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 44.

Claim 53:

The rejection of claim 48 is incorporated. Claim 53, which recite(s) the same limitations as those of claim 6, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 53.

13. Claims 21-24, 27-29, and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Warnes in view of Sokolov.

Claim 21:

Warnes discloses *a method for optimizing the representation of a code sequence, comprising:*

determining a frequency of use of a register based on scanning the code sequence to determine a static frequency of operations in the code sequence (e.g., FIG. 1, blocks 102-106, col.8: 56 – col.9: 38)*;*

modifying the static frequency of use of operations by performing a loop analysis to determine an executed frequency of operations in the register for said code sequence (e.g., col.11, Table 3, col.11: 14 - col.12: 61)*;*

tuning an instruction set for assigning a target-code representation for the register, wherein the tuning of the instruction set is based on the frequency of use of the register (e.g., col.11, Table 2, col.10: 51 – col.11: 27)

wherein the tuning of the instruction set is based on the static frequency of operations and the executed frequency of operations for said register (e.g., col.9: 1-19; col.9: 61 – col.10: 27; col.12: 40 – col.13: 16).

Warnes does not explicitly disclose *analyzing multiple time executed loops to determine an executed frequency of operations.*

However, in an analogous art, Sokolov further discloses *analyzing multiple time executed loops to determine an executed frequency of operations* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

Claim 22:

Warnes discloses *the method of claim 21, wherein the representation of a code sequence is a bit symbol representation* (e.g., col.7: 59 – col.8: 54).

Claim 23:

Warnes discloses *the method of claim 21, wherein the instruction set is a variable length instruction set* (e.g., col.10: 17 – col.11: 62).

Claim 24:

Warnes discloses *the method of claim 21, wherein the instruction set is a constant length instruction set* (e.g., col.9: 20 – col.10: 64).

Claim 27:

Warnes discloses *the method of claim 21, further comprising the step of providing a representation of operations frequency, which represents the frequency of operations performed* (e.g., col.13: 3-64).

Claim 28:

Warnes discloses *the method of claim 27, wherein the representation of operation frequency is a frequency distribution* (e.g., col.7: 40 – col.8: 54).

Claim 29:

Warnes discloses *the method of claim 28, wherein the frequency distribution is a histogram* (e.g., col.9: 56 – col.10: 51).

Claim 34:

Warnes does not explicitly disclose *the method of claim 30, wherein the step of determining operation frequency may further comprising determining an executed frequency of operations within multiple times executed loops for the code sequence.*

However, in an analogous art, Sokolov further discloses *determining an executed frequency of operations within multiple times executed loops for the code sequence* (e.g., page 1 and [0066]).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to combine Sokolov's teaching into Srinivasamurthy's teaching. One would have been motivated to do so to optimize Java virtual machines, especially those operating with limited resources as suggested by Sokolov (e.g., [0015]-[0017]).

14. Claims 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Warnes in view of Sokolov and Cohen.

Claim 26:

The rejection of claim 21 is incorporated. Claim 26, which recite(s) the same limitations as those of claim 6, wherein all claimed limitations have been addressed and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 26.

15. Claim 35 is rejected under 35 U.S.C. 103(a) as being unpatentable over Warnes in view of Cohen.

Claim 35:

The rejection of claim 30 is incorporated. Claim 35, which recite(s) the same limitations as those of claim 6, wherein all claimed limitations have been addressed

and/or set forth above. Therefore, as the reference teaches all of the limitations of the above claim(s), it also teaches all of the limitations of claim 35.

Conclusion

16. Applicants' amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

17. Any inquiry concerning this communication should be directed to examiner Thuy Dao (Twee), whose telephone/fax numbers are (571) 272 8570 and (571) 273 8570, respectively. The examiner can normally be reached on every Tuesday, Thursday, and Friday from 6:00AM to 6:00PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached at (571) 272 3695.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273 8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is (571) 272 2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published

applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Thuy Dao/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192